Appendix

A. Proofs of Theoretical Guarantees

A.1. Auxiliary Lemmas

Lemma 1 (Dominance of the candidate-set oracle)**.** *Under Assumption 2, for any $X \in \mathcal{X}$ let*

$$P^\star_{\mathcal{C}}(X) \in \arg\max_{P \in \mathcal{C}(X)} \mu(P, X).$$

Then, for any $P \in \mathcal{C}(X)$,

$$\mu(P^\star_{\mathcal{C}}(X), X) \;\geq\; \mu(P, X).$$

Proof. Fix any X. By Assumption 2, $\mathcal{C}(X)$ is finite and non-empty, hence $\max_{P \in \mathcal{C}(X)} \mu(P, X)$ exists and is attained. By definition of $\arg\max$, $P^\star_{\mathcal{C}}(X)$ attains this maximum, so

$$\mu(P^\star_{\mathcal{C}}(X), X) = \max_{P \in \mathcal{C}(X)} \mu(P, X).$$

Therefore, for any $P \in \mathcal{C}(X)$,

$$\mu(P^\star_{\mathcal{C}}(X), X) \geq \mu(P, X).$$

□

Lemma 2 (Approximate maximization under uniform score error)**.** *Assume Assumption 2 and that equation 6 holds for some ε. Fix any $X \in \mathcal{X}$ and let $\widehat{P}(X) \in \arg\max_{P \in \mathcal{C}(X)} f_\theta(P, X)$, $P^\star_{\mathcal{C}}(X) \in \arg\max_{P \in \mathcal{C}(X)} \mu(P, X)$. Then*

$$\mu(\widehat{P}(X), X) \;\geq\; \mu(P^\star_{\mathcal{C}}(X), X) \;-\; 2\varepsilon.$$

Proof. Fix any X and abbreviate $\widehat{P} = \widehat{P}(X)$ and $P^\star = P^\star_{\mathcal{C}}(X)$. By optimality of $\widehat{P}$ for f_θ,

$$f_\theta(\widehat{P}, X) \geq f_\theta(P^\star, X).$$

If equation 6 holds, then $\mu(\widehat{P}, X) \geq f_\theta(\widehat{P}, X) - \varepsilon$ and $f_\theta(P^\star, X) \geq \mu(P^\star, X) - \varepsilon$. Combining the three inequalities yields

$$\mu(\widehat{P}, X) \geq \mu(P^\star, X) - 2\varepsilon.$$

□

Lemma 3 (Exact selection when the candidate-set gap is large)**.** *Assume Assumption 2 and that equation 6 holds for some ε. Let $g(X)$ be defined in equation 5. If $g(X) > 2\varepsilon$, then $\widehat{P}(X)$ is a maximizer of $\mu(\cdot, X)$ over $\mathcal{C}(X)$, i.e.,*

$$\mu(\widehat{P}(X), X) \;=\; \mu_{(1)}(X).$$

Proof. Fix any X and let $P_{(1)} \in \arg\max_{P \in \mathcal{C}(X)} \mu(P, X)$ be a maximizer. For any P with $\mu(P, X) \leq \mu_{(2)}(X)$, equation 6 implies $f_\theta(P_{(1)}, X) \geq \mu_{(1)}(X) - \varepsilon$ and $f_\theta(P, X) \leq \mu(P, X) + \varepsilon \leq \mu_{(2)}(X) + \varepsilon$. If $g(X) > 2\varepsilon$, then

$$\mu_{(1)}(X) - \varepsilon > \mu_{(2)}(X) + \varepsilon,$$

hence

$$f_\theta(P_{(1)}, X) > f_\theta(P, X)$$

for all such P. Thus any maximizer of $f_\theta(\cdot, X)$ must also be a maximizer of $\mu(\cdot, X)$, and therefore

$$\mu(\widehat{P}(X), X) = \mu_{(1)}(X).$$

□

Lemma 4 (Expected regret bound under Tsybakov/no-tie)**.** *Assume Assumptions 2 and 4, and that equation 6 holds for some ε. Let*

$$r(X) \triangleq \mu(P^\star_{\mathcal{C}}(X), X) - \mu(\widehat{P}(X), X) \;\geq\; 0.$$

Then

$$\mathbb{E}_{X \sim \mathcal{D}_X}[r(X)] \;\leq\; C_\alpha\, \varepsilon^{\alpha+1}, \quad \textit{where } C_\alpha = 2^{\alpha+1} c_0.$$

Proof. By Lemma 2, $r(X) \leq 2\varepsilon$ for all X. By Lemma 3, $r(X) = 0$ whenever $g(X) > 2\varepsilon$. Hence

$$r(X) \leq 2\varepsilon\, \mathbf{1}\{g(X) \leq 2\varepsilon\}.$$

Taking expectation and using Assumption 4 with $t = 2\varepsilon$ yields

$$\mathbb{E}[r(X)] \leq 2\varepsilon \cdot \mathbb{P}(g(X) \leq 2\varepsilon) \leq 2\varepsilon \cdot c_0 (2\varepsilon)^\alpha = C_\alpha \varepsilon^{\alpha+1}.$$

□

A.2. Proof of Theorem 1

Proof. Let $\mathcal{E}_\delta$ denote the high-probability event in Assumption 3, i.e.,

$$\mathcal{E}_\delta \triangleq \left\{ \sup_{X \in \mathcal{X}} \sup_{P \in \mathcal{C}(X)} \left| f_\theta(P, X) - \mu(P, X) \right| \leq \varepsilon(N_{\mathrm{tr}}, \delta) \right\}.$$

By Assumption 3, $\mathbb{P}(\mathcal{E}_\delta) \geq 1 - \delta$ over the training data. We prove the desired inequality on the event $\mathcal{E}_\delta$.

Throughout the proof, write $\varepsilon = \varepsilon(N_{\mathrm{tr}}, \delta)$ and let

$$r(X) = \mu(P^\star_{\mathcal{C}}(X), X) - \mu(\widehat{P}(X), X).$$

Then

$$\mathbb{E}[\mu(\widehat{P}(X), X)] \;=\; \mathbb{E}[\mu(P^\star_{\mathcal{C}}(X), X)] \;-\; \mathbb{E}[r(X)]. \quad (9)$$

By Lemma 4, on the event $\mathcal{E}_\delta$,

$$\mathbb{E}[r(X)] \le C_\alpha \varepsilon^{\alpha+1}.$$

By Assumption 2, $P^{(\mathrm{avg})} \in \mathcal{C}(X)$ for all X, hence Lemma 1 implies

$$\mu(P^\star_{\mathcal{C}}(X), X) \ge \mu(P^{(\mathrm{avg})}, X) \qquad \text{for all } X.$$

Combining this with Assumption 5 yields

$$\mu(P^\star_{\mathcal{C}}(X), X) - \mu(P^{(\mathrm{avg})}, X) \;\ge\; \gamma_{\mathrm{avg}} \cdot \mathbf{1}\{X \in \mathcal{S}_{\mathrm{avg}}\}.$$

Taking expectation and using $\mathbb{P}(X \in \mathcal{S}_{\mathrm{avg}}) \ge p_{\mathrm{avg}}$ gives

$$\mathbb{E}[\mu(P^\star_{\mathcal{C}}(X), X)] \;\ge\; \mathbb{E}[\mu(P^{(\mathrm{avg})}, X)] \;+\; p_{\mathrm{avg}}\gamma_{\mathrm{avg}}.$$

Substituting into equation 9 yields

$$\mathbb{E}[\mu(\widehat{P}(X), X)] \;\ge\; \mathbb{E}[\mu(P^{(\mathrm{avg})}, X)] + p_{\mathrm{avg}}\gamma_{\mathrm{avg}} - C_\alpha \varepsilon^{\alpha+1}.$$

Under condition equation 7, the right-hand side is strictly larger than $\mathbb{E}[\mu(P^{(\mathrm{avg})}, X)]$. Therefore, on $\mathcal{E}_\delta$,

$$\mathbb{E}_{X\sim\mathcal{D}_X}[\mu(\widehat{P}(X), X)] > \mathbb{E}_{X\sim\mathcal{D}_X}[\mu(P^{(\mathrm{avg})}, X)].$$

Since $\mathbb{P}(\mathcal{E}_\delta) \ge 1 - \delta$, the theorem follows. □

A.3. Proof of Theorem 2

Proof. Let $\mathcal{E}_\delta$ denote the high-probability event in Assumption 3, i.e.,

$$\mathcal{E}_\delta \triangleq \left\{ \sup_{X\in\mathcal{X}} \sup_{P\in\mathcal{C}(X)} \left| f_\theta(P, X) - \mu(P, X) \right| \le \varepsilon(N_{\mathrm{tr}}, \delta) \right\}.$$

By Assumption 3, $\mathbb{P}(\mathcal{E}_\delta) \ge 1 - \delta$ over the training data. We prove the desired inequality on the event $\mathcal{E}_\delta$.

Throughout the proof, write $\varepsilon = \varepsilon(N_{\mathrm{tr}}, \delta)$ and let

$$r(X) = \mu(P^\star_{\mathcal{C}}(X), X) - \mu(\widehat{P}(X), X).$$

As in equation 9,

$$\mathbb{E}[\mu(\widehat{P}(X), X)] \;=\; \mathbb{E}[\mu(P^\star_{\mathcal{C}}(X), X)] \;-\; \mathbb{E}[r(X)]. \quad (10)$$

By Lemma 4, on the event $\mathcal{E}_\delta$,

$$\mathbb{E}[r(X)] \le C_\alpha \varepsilon^{\alpha+1}.$$

By definition of $i^\dagger$ in Assumption 6,

$$\mathbb{E}[\mu(P^{(\mathrm{bs})}, X)] \;=\; \max_{i\in[N]} \mathbb{E}[\mu(P^{(i)}, X)].$$

By Assumption 2, $P^{(\mathrm{bs})} \in \mathcal{C}(X)$ for all X, so Lemma 1 yields

$$\mu(P^\star_{\mathcal{C}}(X), X) \ge \mu(P^{(\mathrm{bs})}, X) \qquad \text{for all } X.$$

Combining this with Assumption 6 gives

$$\mu(P^\star_{\mathcal{C}}(X), X) - \mu(P^{(\mathrm{bs})}, X) \;\ge\; \gamma_{\mathrm{bs}} \cdot \mathbf{1}\{X \in \mathcal{S}_{\mathrm{bs}}\}.$$

Taking expectation and using $\mathbb{P}(X \in \mathcal{S}_{\mathrm{bs}}) \ge p_{\mathrm{bs}}$ yields

$$\mathbb{E}[\mu(P^\star_{\mathcal{C}}(X), X)] \;\ge\; \mathbb{E}[\mu(P^{(\mathrm{bs})}, X)] \;+\; p_{\mathrm{bs}}\gamma_{\mathrm{bs}}.$$

Substituting into equation 10 yields

$$\mathbb{E}[\mu(\widehat{P}(X), X)] \;\ge\; \mathbb{E}[\mu(P^{(\mathrm{bs})}, X)] \;+\; p_{\mathrm{bs}}\gamma_{\mathrm{bs}} \;-\; C_\alpha \varepsilon^{\alpha+1}.$$

Under condition equation 8, the right-hand side is strictly larger than $\mathbb{E}[\mu(P^{(\mathrm{bs})}, X)]$; since

$$\mathbb{E}[\mu(P^{(\mathrm{bs})}, X)] = \max_{i\in[N]} \mathbb{E}[\mu(P^{(i)}, X)],$$

the claim follows on the event $\mathcal{E}_\delta$. Since $\mathbb{P}(\mathcal{E}_\delta) \ge 1 - \delta$, the theorem follows. □

B. Additional Experimental Details and Results

This appendix complements Section 5. Since the main quantitative benchmark table is reported in the main text, this appendix focuses on the dataset protocol, the full implementation details, the computation cost comparison, and additional qualitative examples. Appendix Table 3 summarizes the dataset protocol. Appendix Table 4 reports the complete configuration of FRAME. Appendix Table 5 reports the resource comparison that complements RQ4. Appendix Figure 3 provides additional qualitative comparisons across representative baselines and FRAME.

B.1. Implementation Details

We train the learned components of FRAME exclusively on the CASIA v2 dataset, which provides 8,831 training images and 1,918 validation images. No test-set images are used during training or model selection. All evaluation is performed on four held-out benchmarks: CASIA v1, Coverage, Columbia, and RealisticTampering. Columbia is used only for image-level detection, since we do not use pixel-level localization masks for that dataset under our protocol. This strict train-test separation ensures that all reported results measure generalization to unseen data distributions rather than reuse of benchmark-specific statistics. Table 3 summarizes the dataset protocol.

Table 3. Dataset protocol used in all experiments. The selector and fusion module are trained only on CASIA v2. All methods are evaluated on the same four test sets.

Role	Dataset	Size	Usage
Train	CASIA v2 train	8,831	Train selector and fusion
Val	CASIA v2 val	1,918	Validation and early stopping
Test	CASIA v1	1,754	Main benchmark
Test	Coverage	200	Copy-move benchmark
Test	Columbia	363	Detection only
Test	RealisticTampering	440	Realistic splicing benchmark

FRAME operates over a fixed pool of handcrafted forensic algorithms from pyIFD [26]. The core pool contains nine modules that cover JPEG compression cues, sensor noise patterns, and camera-related artifacts. We also include six optional modules in the main experiments, which gives a total pool of fifteen modules. All module outputs are normalized heatmaps in $[0, 1]$. These outputs are precomputed offline and cached to disk before any model training or evaluation. This design ensures that the learned components operate on fixed forensic evidence and do not repeatedly execute the handcrafted algorithms during training.

A path is an ordered sequence of one to four modules drawn from the module pool. For each image, we sample $K = 50$ candidate paths, with up to $\max(10K, 100) = 600$ attempts to avoid duplicates. Module outputs within a path are fused by uniform averaging to produce a path-level heatmap. The GNN selector then scores all sampled paths, and the top-$k = 5$ paths are retained for final fusion. The selector is a lightweight GraphSAGE-style network [13] with three message-passing layers, each with hidden dimension 64. Each module is represented as a graph node with a learnable 64-dimensional embedding. Mean pooling over node embeddings is concatenated with a 9-dimensional image feature vector and a 5-dimensional manipulation-type one-hot vector, and the resulting representation is processed by a small MLP to produce a scalar path-quality score. The selector itself is lightweight and contains approximately 44,000 parameters.

The image-level features include log-scaled height and width, mean and standard deviation of grayscale intensity, Shannon entropy, Canny edge density, saturation ratio, and binary indicators for JPEG and PNG formats. The selector is trained by regression, where the target is the pixel-level F1 score of the path-level heatmap against the ground-truth mask. We use Adam with learning rate 10^{-3}, weight decay 10^{-4}, batch size 128, and 15 training epochs, and we retain the checkpoint with the lowest validation loss. The fusion module learns a scalar weight for each selected path. These weights are optimized using a combined BCE and Dice loss with equal coefficients. During fusion training, images are downsampled to a maximum side length of 384 pixels. At inference time, the top-k selected heatmaps are combined by learned softmax weighting with temperature $\tau = 1.0$. The final image-level detection score is the maximum pixel value of the fused heatmap. Pixel-level localization is computed by thresholding the fused map at 0.5. Table 4 summarizes the full configuration.

B.2. Computational Cost Analysis

Table 5 reports the resource comparison that complements RQ4. The table shows a clear trade-off between modularity and wall-clock efficiency. The deep learning baselines are substantially faster at inference, which is expected because they execute a single learned model on GPU. In contrast, FRAME evaluates multiple handcrafted forensic modules, many of which are CPU-bound. This design makes the overall runtime much higher than the deep learning baselines. However, this cost should be interpreted together with the modular design of the method. The runtime is dominated by handcrafted module execution rather than by the selector or the learned fusion module.

At the same time, the learned part of FRAME is lightweight. It uses much less GPU memory than the deep learning baselines and maintains a very small learned parameter footprint. The storage cost is also moderate. Compared with uniform-all pyIFD, FRAME increases runtime only modestly while providing much stronger accuracy. This result suggests that the main cost comes from the shared forensic module pool rather than from the learned selection and fusion mechanism. Overall, the numbers in Table 5 show that FRAME trades inference speed for modularity and stronger localization, while keeping the learned component small.

B.3. Additional Qualitative Results

Figure 3 provides additional qualitative comparisons between representative deep learning baselines, representative handcrafted cues, and FRAME. Several examples show that DCT and other single handcrafted cues can respond strongly but also produce noisy background activation. TruFor and MMFusion are generally more spatially consistent than the single handcrafted cues, yet their responses may still miss fine boundaries or weaker manipulated regions. FRAME usually produces a more concentrated response in the manipulated area while suppressing part of the spurious background activation that appears in the handcrafted maps.

The figure also shows that the problem remains difficult in some low-texture or small-object cases. In such examples, all methods weaken, and the advantage of FRAME becomes smaller. Still, the qualitative comparisons support the main conclusion of the paper: adaptive selection and learned fusion help combine complementary forensic signals into a cleaner final localization map.

Table 4. Complete experimental configuration of FRAME.

Category	Parameter	Value	Description
Dataset	Train set	CASIA v2	8,831 train / 1,918 val images
	Test sets	CASIA v1, Coverage, Columbia, RealisticTampering	External benchmarks only
	Mask binarization thr	0.5	GT and predicted mask threshold
	Detection score	max pixel	Max of fused heatmap
	Random seed	0	All sampling and evaluation
pyIFD Pool	Core modules	9	ELA, DCT, NOI1–5, GHOST, BLK, CAGI
	Optional modules	6	ADQ1–3, NADQ, CFA1, CAGI_INV
	Total pool	15	All used in main experiments
	Module output	heatmap $\in [0, 1]$	Per-pixel anomaly score
Path Sampling	Candidate paths K	50	Default; varied in sensitivity study
	Min / max path length	1 / 4	Modules per path
	Max graph nodes	8	Supernet nodes per sample
	Max sampling attempts	$\max(10K,\ 100)$	600 for $K{=}50$
	Intra-path fusion	uniform	Equal weight within a path
GNN Selector	Architecture	GraphSAGE, 3 layers	Message-passing GNN
	Hidden dimension	64	All GNN layers
	Module embedding dim	64	Learnable module ID embedding
	Image feature dim	9	Hand-crafted features
	Manip. type dim	5	One-hot encoded
	MLP head	$78 \to 128 \to 64 \to 1$	ReLU + sigmoid output
	Selector parameters	≈44,000	Checkpoint < 0.2 MB
Image Features	Spatial	$\log(1{+}H)$, $\log(1{+}W)$	Log-scaled resolution
	Intensity	mean / 255, std / 255	Grayscale statistics
	Texture	entropy / 8, edge density	Canny (100 / 200)
	Clipping	saturation ratio	Pixels ≤ 2 or ≥ 253
	Format	is_JPEG, is_PNG	Binary codec flags
GNN Training	Optimizer	Adam ($\beta_1{=}0.9,\ \beta_2{=}0.999$)	—
	Learning rate	10^{-3}	Fixed schedule
	Weight decay	10^{-4}	L2 regularization
	Batch size	128 paths	—
	Epochs	15	Best val-loss checkpoint saved
	Gradient clipping	5.0 (global norm)	—
Fusion Training	Optimizer	Adam	Same betas as GNN
	Learning rate	10^{-2}	Fixed schedule
	Weight decay	10^{-4}	—
	Epochs	10	Fixed schedule
	Loss	BCE + Dice, $\lambda{=}1.0$ each	Dice $\epsilon{=}10^{-6}$
Inference	Fused paths k	5	Top-k from GNN; varied in study
	Fusion strategy	learned weights	Trained on CASIA v2
	Softmax temperature	$\tau{=}1.0$	Score-softmax weighting
	Max image side (train)	384 px	Downsampled during fusion training only
Hardware	GPU	NVIDIA H200 (80 GB)	GNN / fusion training and inference
	CPU workers	16	pyIFD precomputation
	RAM	128 GB	—

Table 5. Computation cost comparison measured on CASIA v1. FRAME runs pyIFD modules on CPU and the GNN selector on GPU. Wall time is dominated by CPU-bound module execution. †: no learned parameters.

Method	Time (s/img)	GPU Mem (MB)	CPU Mem (MB)	Params (M)	Model Size (MB)	Storage (MB/img)
Best single pyIFD	1.12	0	184	$0^{\dagger}$	$0^{\dagger}$	0.18
Uniform-all pyIFD	8.42	0	312	$0^{\dagger}$	$0^{\dagger}$	1.46
TruFor	0.34	2,847	891	149.6	599.2	0.12
MMFusion	0.28	3,124	746	83.2	332.8	0.09
ManTraNet	0.19	1,892	634	38.4	153.6	0.08
CAT-Net	0.41	2,156	712	56.7	226.8	0.10
FRAME (ours)	9.23	847	428	0.04	0.2	0.24

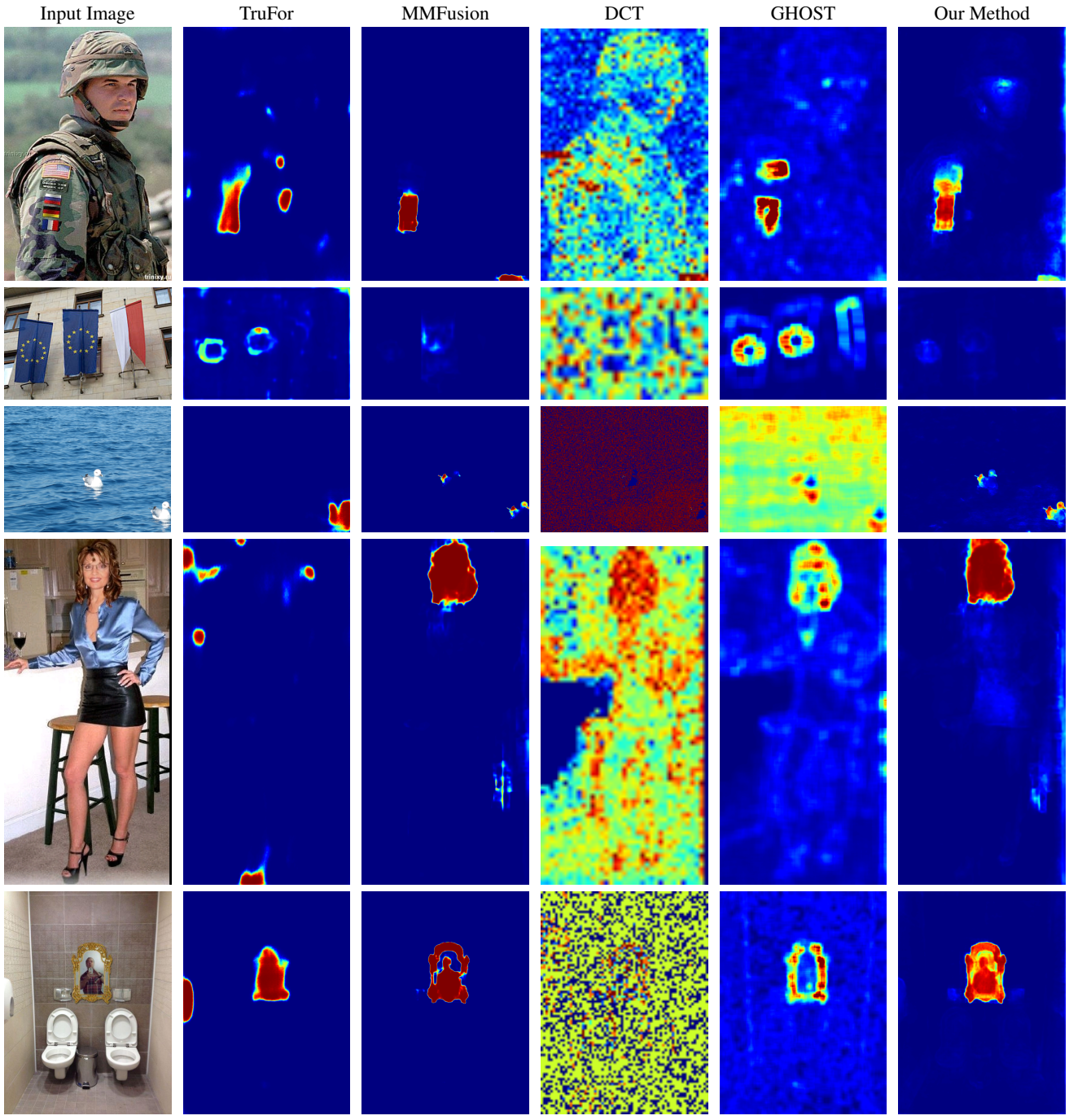

Figure 3. Additional qualitative comparison of manipulation localization on five examples. Columns from left to right show the input image, TruFor, MMFusion, DCT, GHOST, and FRAME. The examples illustrate how FRAME combines complementary forensic signals and often produces a cleaner response around the manipulated region.